



Tech status and directions



Status and directions

- We try to summarize here the items that
 - have characterized the past year
 - Suggest a direction for the future steps
- Main aspects that are considered
 - Consolidation
 - Making DPM better, privilege solidity and simplicity, reduce the complexity and the effort
 - Evolution
 - Adapt to a changing landscape of tools/libs/etc
 - Adapt to the evolving requirements of users
 - Go towards user-friendly interfaces
 - Exploration
 - Explore features that will likely be needed next
 - Try to see if a new thing can be a possible evolution
 - Clouds, federations, exotic plugins fall here

1.8.7 is **the EPEL DPM**

- Component-based releases is among the decisions of the past years
- DPM is composed by components that are packaged and released independently
 - Each one has its own version number
 - The glorious LCG-DM component is the only one that has version 1.8.7 now (next will be 1.8.8)

1.8.7 is **the EPEL DPM**

- Big improvements to the puppet configuration
- DB Connection pooling in the DPNS/LFC daemon
 - Improves performance 3X-5X and reduces famous hiccups
- The EGI/OGF Storage Accounting Record producer (StAR)
 - Added to the EPEL rpm since a few days
- Lots of bug fixes
- Brand new Xrootd interface using native DMLite (good feedback on the prev one, from ALICE and ATLAS FAX)
- Brand new GridFTP2 with redirections is coming
- Dynamic HTTP Federations (implemented as a dmlite plugin called “UGR”, only in RC by now)

What is a DPM release

- We mentioned releases of components
- We welcome people to join the development and the testing
- We work with a continuous build system that keeps always up to date two work repositories, available to the public
- These are always supposed to be installable and work
 - Trunk repo
 - <http://grid-deployment.web.cern.ch/grid-deployment/dms/lcgdm/repos/>
 - Release Candidate repo:
 - <http://grid-deployment.web.cern.ch/grid-deployment/dms/lcgdm/repos/release-candidate/>
- The next slides describe the process and show where collaborators' activities can be plugged

Architectures

- We have 32 components
- We have 2 archs (i386 and x86_64)
- We have 2 platforms (el5, el6)
 - Minus el5-i386, which we don't support
- Hence, ± 100 RPMs, as some components produce more than one
- This process started ± 2 years ago, when we incrementally started using 'mock' to build
- The build management overhead of such a mass can be overkill for us humans, a detail out of place can trigger the need for a lot of manpower just to fix little configurations
- We prefer to concentrate on the quality than on the overhead, hence... we want VERY reliable builds, babysitting is out of question

Our continuous builds now

- Minimalistic approach
 - The devs develop
 - The build system builds (what the devs do)
 - The nightly tests ... deploy and test (what's built)
 - The testbeds self update and run (what's blessed)
- We remove all the config bits that are redundant for this purpose
- Master rule: Never have to touch the build system after changes in the code.
 - The build system just ... builds!

Repositories we provide

- Master rule: everything has always to compile and install correctly
- The repositories are OPEN, everyone can use them to install a DPM
- **Trunk:** where we work and add features. Things may be slightly incomplete here.
- **Release-candidate:** the set of the component versions that have been blessed as “well working” by the developers
- When we are happy, we proceed with the EPEL workflow

Continuous *Trunk* repo

- We build in a continuous way the trunk of all the components using epel-testing as external package provider
- Populate 3 YUM repos (2 archs, 2 platforms minus el5_i386) with the rpm and srpm artifacts
- Make these repos available for
 - The developer, with a low latency (minutes)
 - The nightly deployment and functional tests
 - *yum* like a normal repository would
- Our trunk testbed selfupdates every day

Continuous Candidate Release repo

- Minimalistic approach
- The set of packages of the component versions that have been blessed by the devs
 - The responsible dev feels comfortable with users using his component
 - The responsible dev “marks” a tag as RC using an “*svn:externals*” link
 - A directory with a known name (**candidate-release** :-) points to the blessed tag, like a symlink
 - The build system ... builds whatever is named **candidate-release** every time its content changes
 - No parameters, hence no tweaks are needed
- Our RC testbed selfupdates every day



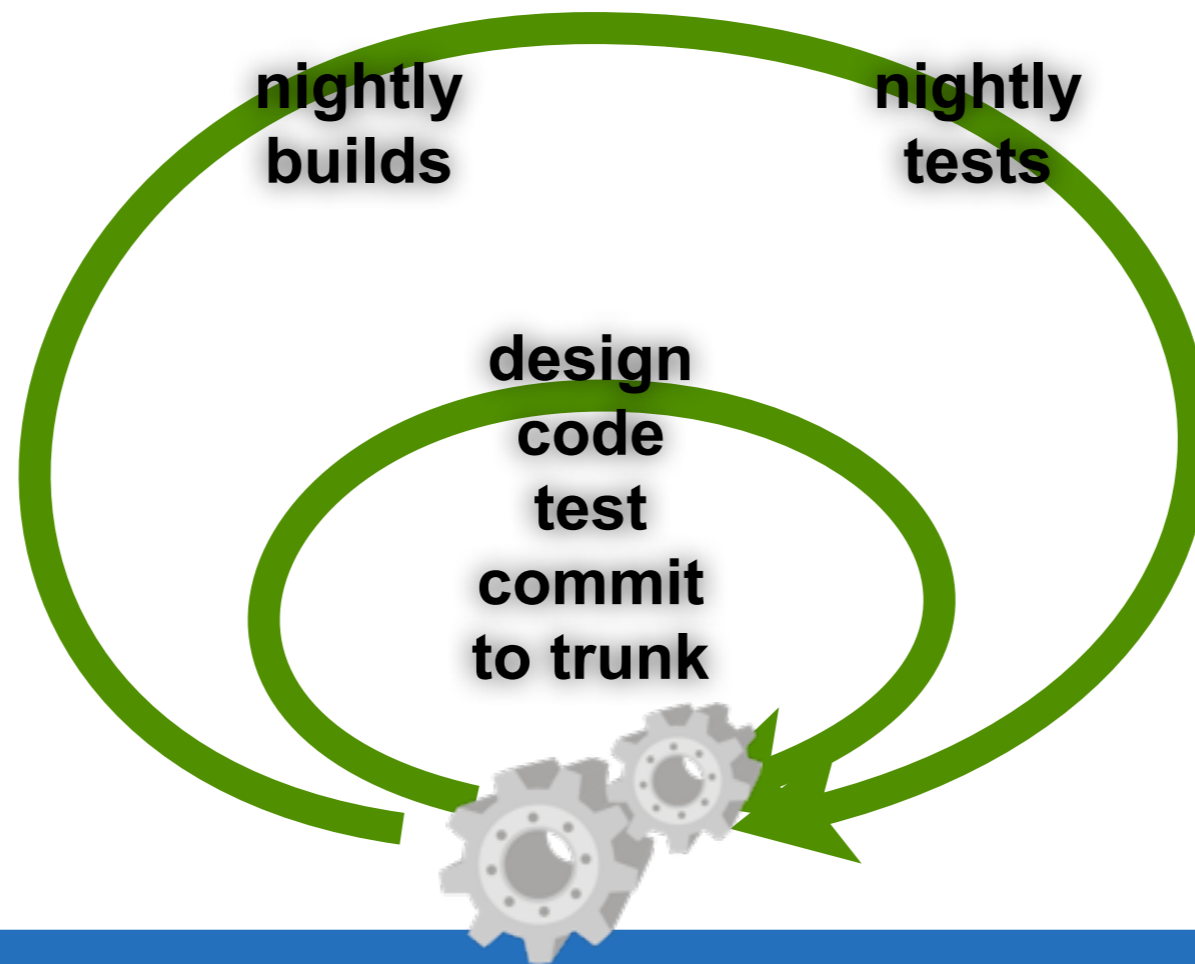
IT-SDC

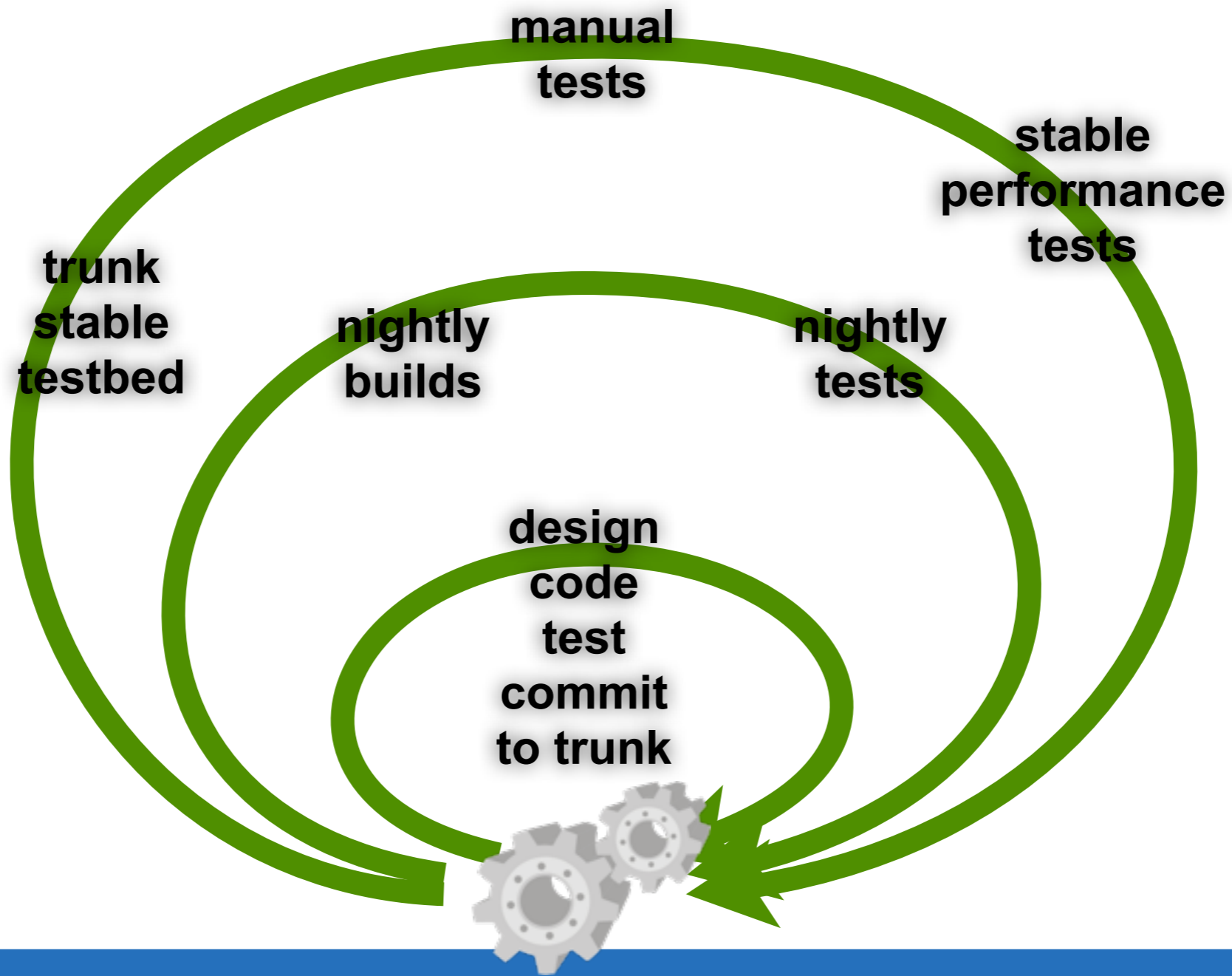
DPM Workshop

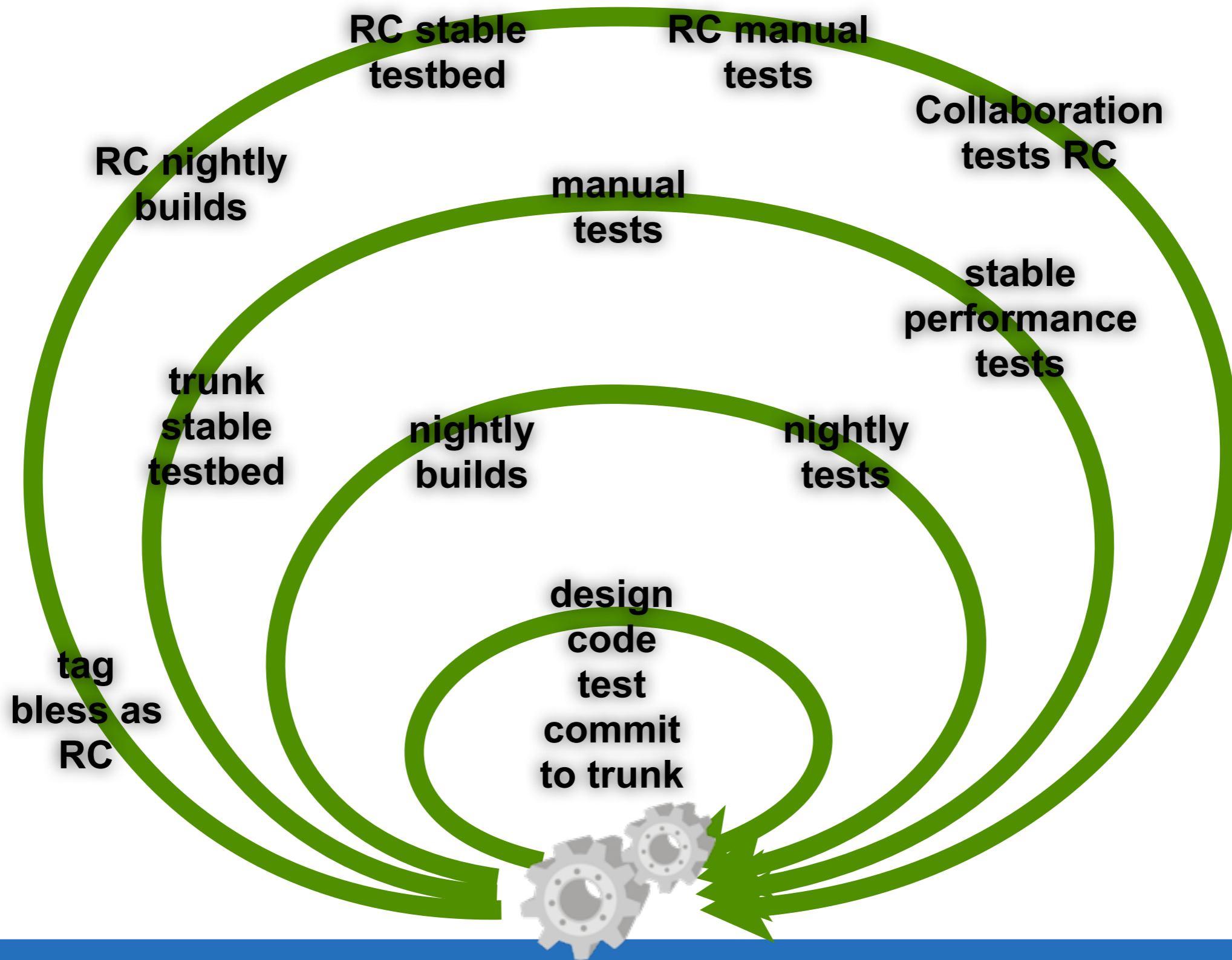
13 Dec 2013

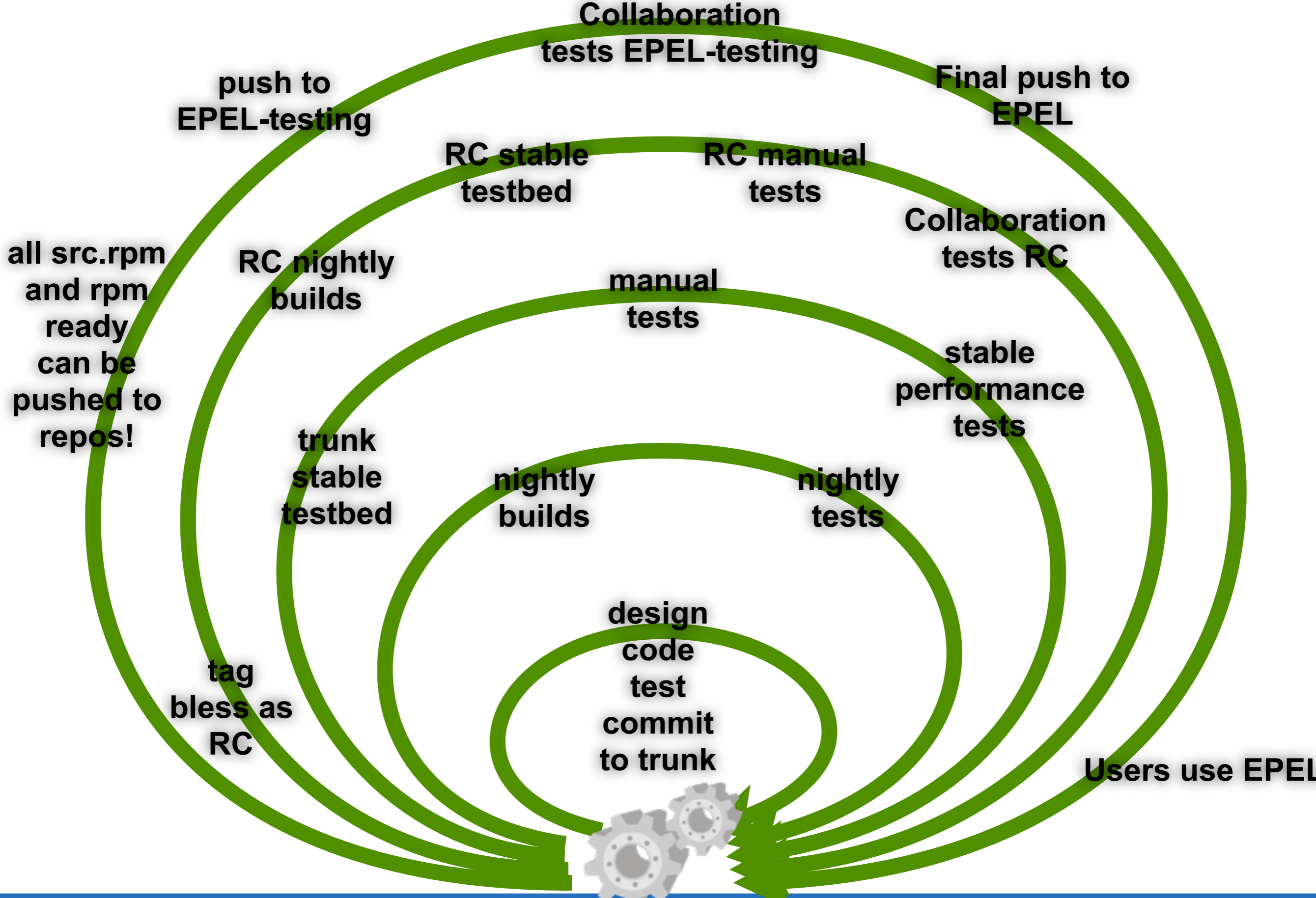
**design
code
test
commit
to trunk**

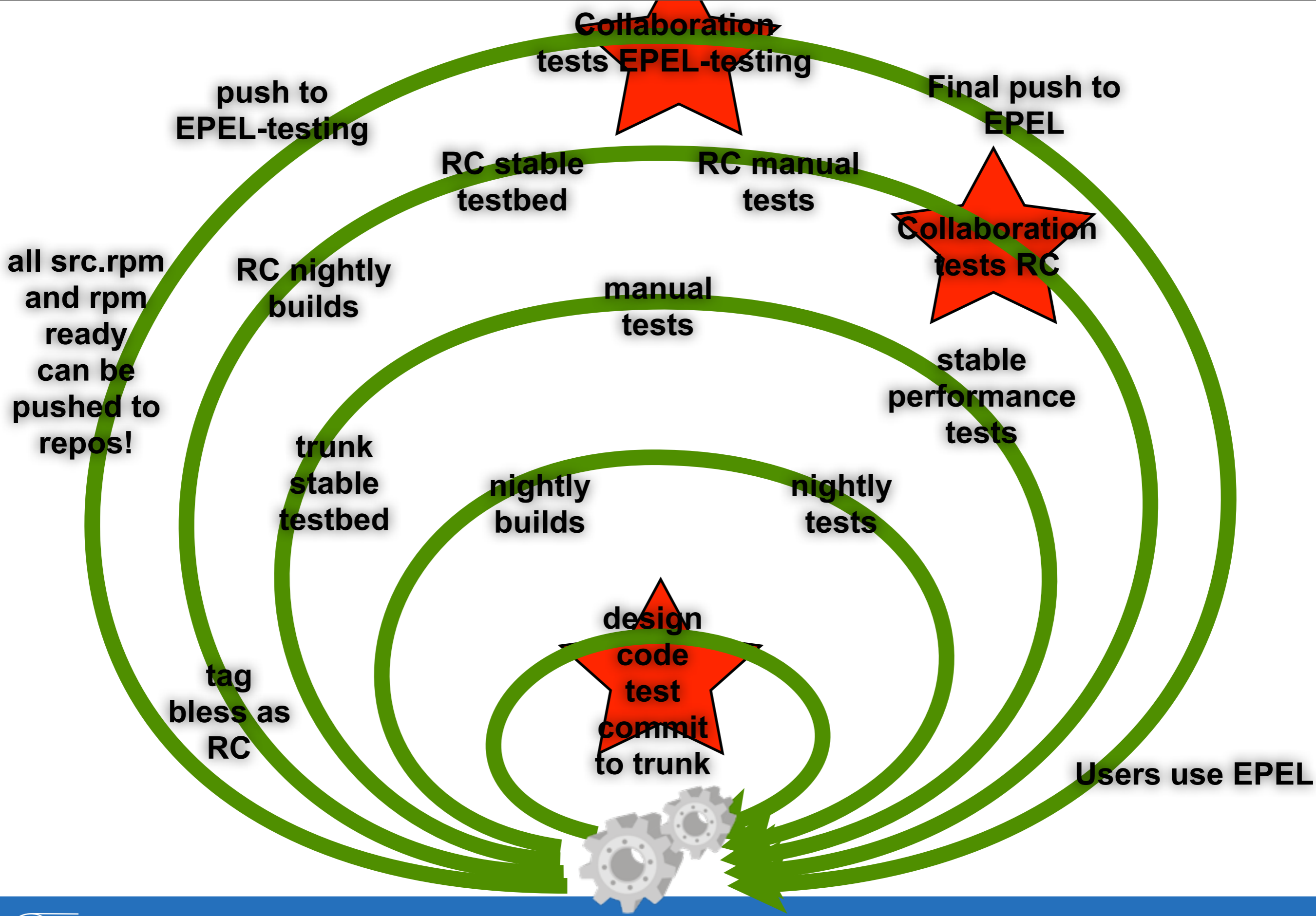














IT-SDC

DPM Workshop

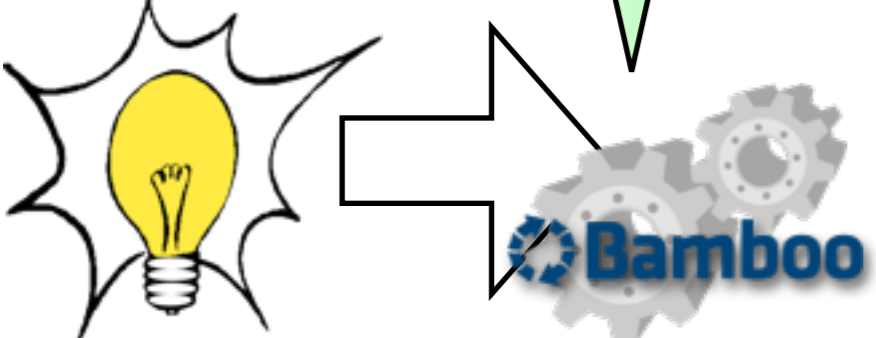
13 Dec 2013

**design
code
commit
to trunk**



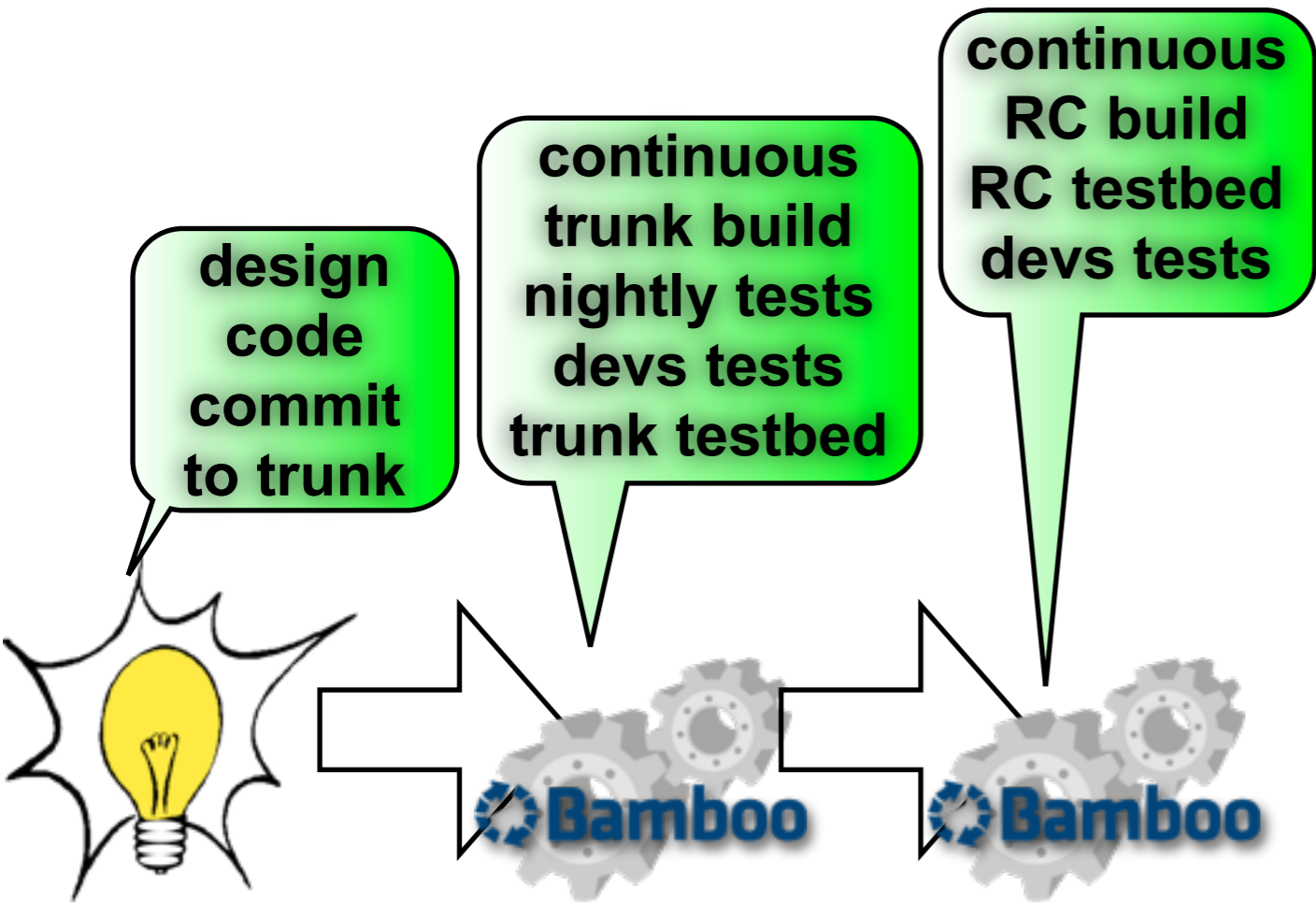
design
code
commit
to trunk

continuous
trunk build
nightly tests
devs tests
trunk testbed



trunk
YUM repo

Trunk
stable testbed



design
code
commit
to trunk

continuous
trunk build
nightly tests
devs tests
trunk testbed

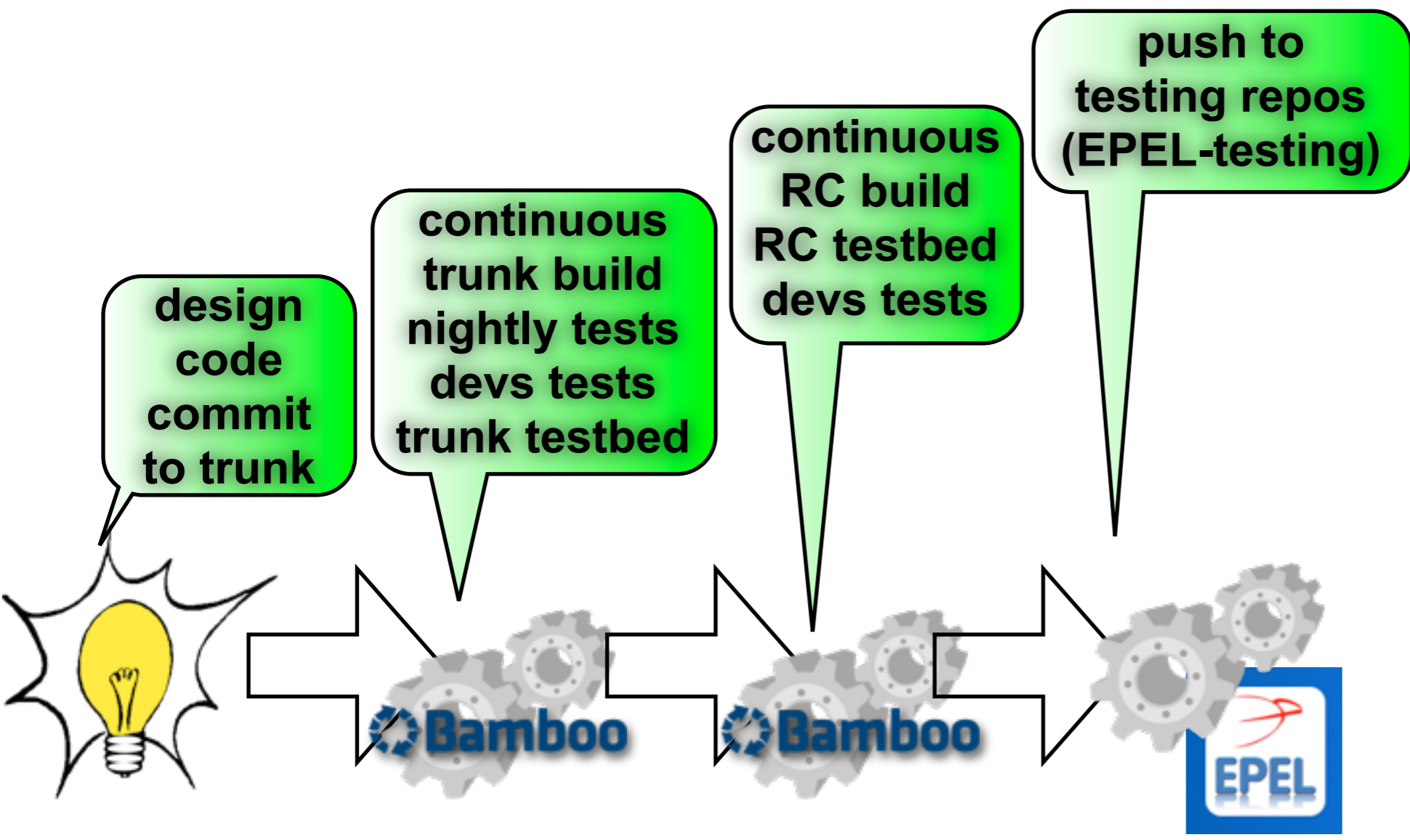
continuous
RC build
RC testbed
devs tests

trunk
YUM repo

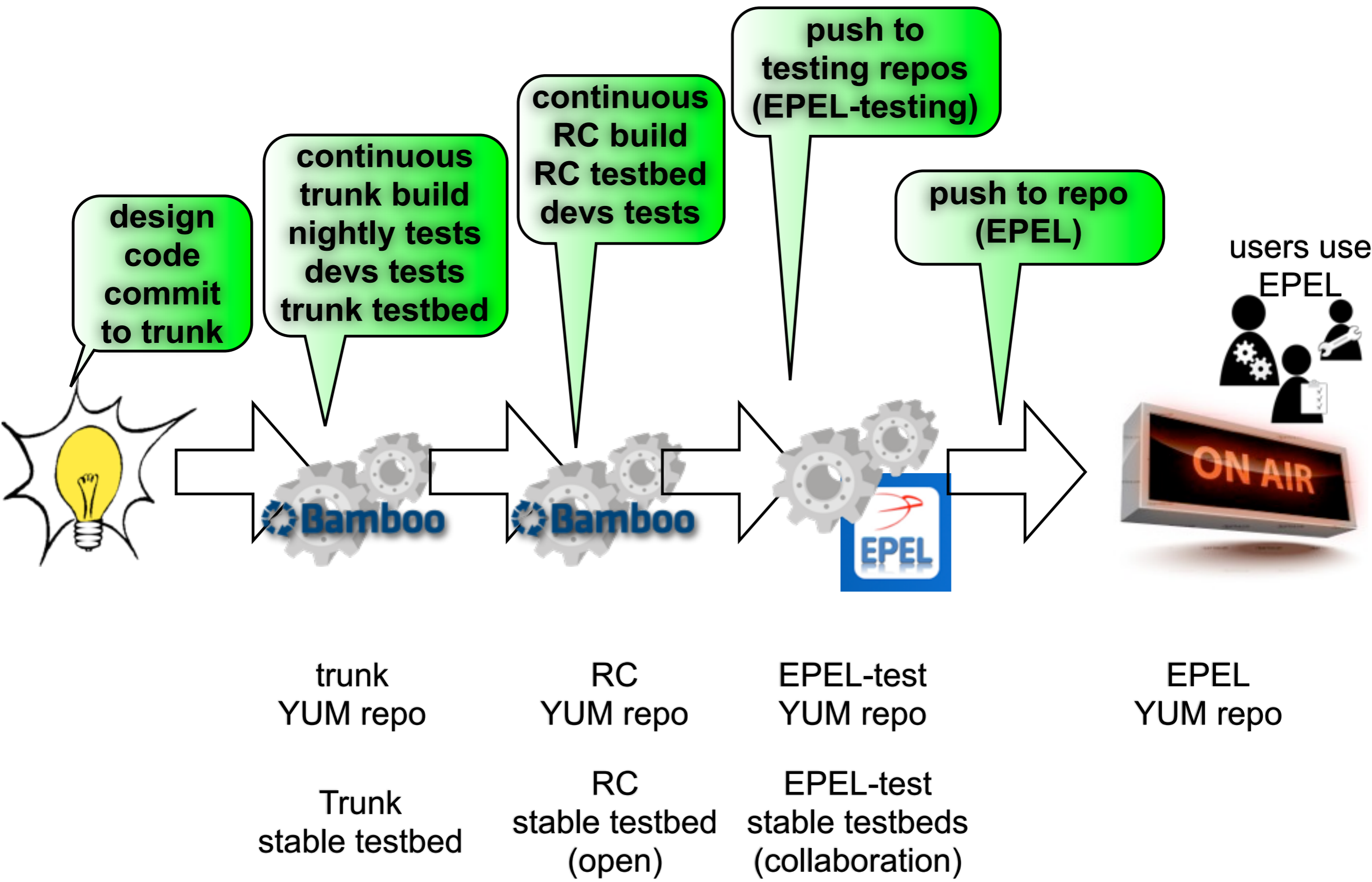
Trunk
stable testbed

RC
YUM repo

RC
stable testbed
(open)



trunk YUM repo	RC YUM repo	EPEL-test YUM repo
Trunk stable testbed	RC stable testbed (open)	EPEL-test stable testbeds (collaboration)



The new visualization

how	FTS3				
how	gCube				
ide	lcmd				
		bamboo-automate			1/1 Successful
		bamboo-automate	2013-08-22 09:03:16	8 minutes	#13 Successful
		cloudera-hadoop			5/5 Successful
		cloudera-hadoop	2013-07-22 09:35:22	45 minutes	#78 Successful
		release-candidate SL5 x86_64	2013-07-22 11:39:31	39 minutes	#7 Successful
		release-candidate SL6 i386	2013-07-22 14:08:26	49 minutes	#5 Successful
		release-candidate SL6 x86_64	2013-07-22 10:32:27	48 minutes	#3 Successful
		trunk SL5 x86_64	2013-07-22 12:47:41	79 minutes	#3 Successful
		dmlite			6/6 Successful
		Build trunk SL6 x86_64	2013-09-25 15:02:53	13 minutes	#459 Successful
		release-candidate SL5 i386	2013-07-03 13:28:43	10 minutes	#9 Successful
		release-candidate SL5 x86_64	2013-09-25 15:03:40	10 minutes	#7 Successful
		release-candidate SL6 i386	2013-09-25 15:03:40	11 minutes	#13 Successful
		release-candidate SL6 x86_64	2013-09-25 15:03:42	10 minutes	#25 Successful
		trunk SL5 x86_64	2013-09-25 15:03:48	8 minutes	#39 Successful
		dmlite-plugins-adapter			6/6 Successful
		Build trunk SL6 x86_64	2013-09-11 22:08:44	10 minutes	#192 Successful
		release-candidate SL5 i386	2013-07-10 10:32:04	13 minutes	#7 Successful
		release-candidate SL5 x86_64	2013-07-10 11:09:01	17 minutes	#25 Successful
		release-candidate SL6 i386	2013-07-10 10:37:41	11 minutes	#13 Successful
		release-candidate SL6 x86_64	2013-07-10 10:42:52	20 minutes	#11 Successful
		trunk SL5 x86_64	2013-09-11 22:06:00	7 minutes	#43 Successful
		dmlite-plugins-hdfs			5/5 Successful
		Build trunk SL6 x86_64	2013-08-19 15:20:44	13 minutes	#114 Successful
		release-candidate SL5 i386	-	-	No completed build
		release-candidate SL5 x86_64	2013-07-03 13:37:13	44 minutes	#4 Successful
		release-candidate SL6 i386	2013-07-03 14:34:47	21 minutes	#4 Successful



YAIM deprecated

- We will maintain YAIM until August 2014
- Its successor is a Puppet-based setup, totally mainstream tools
- We encourage the members of the collaboration to learn the new setup technology and give feedback
 - Now it's the right moment

The HTTP Ecosystem

- We refer as “HTTP Ecosystem” to a sort of critical mass of services that opens possibilities that are flexible, powerful and “industry standard”
- **DPM and the other SEs** provide now many features through HTTP/WebDAV
 - They are in production in many sites, ATLAS uses them
- **LFC** has an HTTP/WebDAV native interface
- **The Dynamic Federations** can federate this kind of things, including opportunistic storage and cloud storage
- **DAVIX** implements in one client the HTTP/WebDAV features that are normally sparse in many different clients
- This has to do with deploying DPM testbeds, as they can be federated easily. We now have all the relevant pieces.

EMI->DPM Collaboration

- The DPM Collaboration has been formed to support the ongoing development and maintenance of DPM.
- Started 2nd May 2013.
- Collaboration Meeting on 23rd April, 2013 with representatives from :
 - CERN
 - Czech Rep.
 - France
 - Italy
 - Japan
 - Taiwan
 - UK

What's in the collaboration (From April)

- Approval of a Collaboration Agreement
- Pledges of effort (over 4FTE)
- In return for influence on direction
- Approval of distribution of responsibility
- Development, testing, support inc all derived and related services
 - LFC and Dynamic Federations
 - HTTP/DAV, NFS, xrootd, gridftp frontends
 - S3, HDFS, VFS backends
 - Memcached for the nameserver
 - Some new developments (eg cluster FS support)
- Best effort support will be provided to non-member communities

Partner responsibilities (From April)

- To give a flavour of tasks (incomplete list)
 - ASGC: NFSv4.1, performance testing
 - Czech: ClusterFS support, GSSAPI/X509
 - France: User support, testing, admin interface
 - Italy: Testing, federation
 - Japan: Testing
 - UK: Admin tools, support, performance evaluations

How can one contribute to the DPM collab

- Contribution: Run a Testbed
 - Install one with your preferred features
 - Make sure that you USE it. Can be Hammercloud tests, whatever
 - Raise your hand if anything breaks after an update of any component
 - HC tests
 - From RC: May break sometimes, I think it happened once, lasted a few days
 - From EPEL-test: should never break! Important to doublecheck!
 - Think about a regional federation built on top of these (e.g. Italy)
 - Make it available to Wahid's HC tests
- Contribution: develop new things
 - features: things that the system can do
 - components: new parts of the system
 - interfaces: ways to interact with the system

HC tests

- With Wahid we are setting up a stable test infrastructure, visible from here:
- <http://ivukotic.web.cern.ch/ivukotic/DPM/index.asp>
- Very promising and interesting results
- Running basic tests that
 - Exercise the advanced I/O optimization features
 - Are debuggable --> makes improvement possible
- Work in progress

Frontend/Backend status

- Xrootd
 - Reports on the next slides
- GridFTP
 - See talk by Andrey
- HTTP/DAV
 - See talk by Alejandro
- NFS
 - See talks by Shu-Ting and Marcel
- HDFS, S3
 - See talk by Martin and Andrea
- RFIO
 - We will try to start building client tools that are not based on it

DPM-xrootd: Current release

- Current stable release is 3.3.4
 - Released in September
 - Available in EPEL 5 and 6
 - XRootD 3.3.4 is in epel
- What's new?
 - Many parts rewritten to use dmlite
 - Support voms extensions via vomsxrd
 - Support queryspace (needed by, e.g. xrootdfs)
 - Support for XRootD 3rd party copy

Courtesy of David Smith

DPM-xrootd: Support for vomsxrd

- vomsxrd is a standard XRootD plugin to support voms
- dpm-xrootd can pass the information to DPM for authorization decisions and XRootD will use it to tag the user's vo in monitoring data
- vomsxrd is in the EMI third party repository
- Installation is optional (dpm-yaim automatically configures vomsxrd only if it is installed)
- Not mandatory to install because disk nodes must be upgraded before vomsxrd is installed on the head node

Courtesy of David Smith

DPM-xrootd: Currently known issues

- Hang of redirectors; crash of redirectors; two problems believed understood, both having workarounds tested.
- queryspace fails if there is a pool defined with no filesystems; will be fixed in a future dmlite release
- Failure to redirect upwards in a federation if the dpm pool containing a file is unavailable; fix in next dpm-xrootd release.

Courtesy of David Smith



DPM-xrootd: Development plans

- Next release expected to be 3.3.5
 - Has improved redirection on failure
 - Resume using `sendfile()` when possible
- Development plans (for 3.3.5 or later)
 - To be able to return file checksum
 - Recompile and port to XRootD 4 once it is released

Courtesy of David Smith

DPM: General Direction highlights (1/2)

- If there is an access/management feature to add, HTTP/DAV will be the interface
- Performance testing/WAN
 - Full steam on the Collaboration testbeds and federations
 - Contribute to Wahid's HC tests, use them as a summary of LAN/WAN access to Collaboration testbeds
- Standardize the Puppet deployment
 - Get production experience
- Direct file placement. Make the primitives available
 - Evaluate it with Sam Skipsey, for the rebalancing scripts
- General focus on interfaces
 - Admin interfaces (shell, HTML/Java interfaces?)
 - Client tools (e.g. dpm-drain) that can work without rfio and potentially give more user-friendly interfaces

DPM: General Direction highlights (2/2)

- Consolidate S3 and HDFS backends, go towards production
- Acquire new frontends and backends. VFS and NFS are first in the list.
- Allow multiple dmlite configurations for different frontends
 - Separate the ones that need the tunnelling
 - Shrink the need for the internal tunnelling
 - Get production experience from GridFTP2, using redirections
- Explore HTTP as internal tunnelling protocol
 - Aim: substitute the rfiio calls in the DMLite adapter plugin
 - Use rfiio only for clients explicitly willing to use rfiio
- Help others in setting up HTTP federations using the Dynamic Federations
 - (DMLite + UGR federator + DPMs)

Conclusion

- 45PB is a respectable size
- The support load seems to be calm and steady, good sign
- Healthy collaboration, healthy discussions in the forums
- Intense development activity
 - Fixing
 - Proactive new features
- Participating to the same Hammercloud tests run by ATLAS



Thank you

